REMARKS

Claims 1-19 will be pending upon entry of the present amendment. Claim 17 is being amended. No new matter is being presented.

The instant invention aims to produce a microprocessor with an extended memory array that can execute the same programs as a microprocessor with a smaller memory array. One embodiment of the instant invention provides for a method of saving and restoring program contextual data to a microprocessor stack during a context switch. The contextual data contains a program counter extended (PCE) addressing register that will contain data if a program is utilizing the extended memory array. During a save of contextual data, the PCE register is only saved to a stack if it contains data. If the PCE has been saved, a flag is set in the condition code register (CCR). The CCR is part of the program contextual data, so the flag itself is saved to the stack during a context save. During a context restore, the CCR and the flag are restored from the stack and the value of the flag indicates whether the PCE is to be restored as well. This process ensures that the PCE is saved or restored during a context switch only if it contains data, and provides for an efficient program context switch on a microprocessor running programs that may utilize an extended memory array.

Claims 1-2, 5-10, 12-17 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tremblay et al (U.S. Patent Application Publication 2001/0010075) in view of admitted prior art.

Tremblay describes a processor with a large register file. A "dirty bit storage" holds information on which registers have been written to since a program was loaded or since the last context switch. Tremblay, par. 20. During a context save, only those registers that have been written to are saved. Tremblay, par. 20. The "dirty bit storage" is initialized when the new program context is loaded. Tremblay, par. 20.

Tremblay does not teach the invention in claim 1, which recites "saving contextual data ... in a variable number of registers that varies according to the value of at least one flag stored in a register to be saved." Though Tremblay does teach saving only those registers indicated by a "dirty bit storage" during a context save, Tremblay par. 70-77, his

invention specifies the "dirty bit storage" as an area in memory separate from the program context data. Tremblay, figures 5 and 6. Tremblay does not state that his "dirty bit storage" is saved as part of the program context. In this way, his invention differs from the invention in claim 1 which specifies at least one flag in a register "to be saved" as part of the program contextual data. With this distinction, the invention of claim 1 is non-obvious in view of Tremblay and the admitted prior art.

The applicants respectfully disagree with the Examiner's statement that Tremblay's "history dirty register" holds flags that signify which registers are to be saved before a context change. This "history dirty register" is a back-up of a register containing program contextual data being written to during program execution. Tremblay, par. 106. It is not a back-up of the "dirty bit storage" and thus does not hold the "dirty bit" flags that indicate which registers are to be saved.

Claims 2 and 5-8 are dependent on claim 1 and are thus also non-obvious for the reasons given above.

Claim 9 is non-obvious for the same reason as claim 1 in the discussion above. Claims 10 and 12-16 are dependent on claim 9 and are thus also non-obvious.

There are additional grounds for asserting the non-obviousness of dependent claim 6. Tremblay does not teach that claim's invention, which recites "restoring the contextual data contained in a variable number of registers that varies according to the value of the flag present in the restored register." Tremblay's "dirty bit storage" is only used to determine which registers have changed since the last context save and so need to be re-saved before the next context change. Tremblay, par. 72. It is not used to determine which registers are restored from memory when the context is restored. In fact, Tremblay's "dirty bit storage" is initialized after each context change, so its information is lost and cannot be used for context restore. Tremblay, par. 72. Given this fact, Tremblay must restore the entire context, not a "variable number of registers." The invention of claim 6 teaches a flag that is used on context restore as well as context save, so claim 6 is non-obvious in view of Tremblay and the admitted prior art.

Claim 14 is non-obvious for the same reasons as given for claim 6.

Claim 17 has been amended to state that the flag is stored in a register to be saved as part of the program contextual data. With this amendment, claim 17 is non-obvious in view of Tremblay and the admitted prior art for the reasons expressed above with respect to claim 1.

Claim 19 is dependent on claim 17, and is thus non-obvious for the same reasons as claim 17.

Claims 3-4, 11-12 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over admitted prior art and Tremblay, further in view of Kishida et al (U.S.P.N. 6,199,155).

Kishida describes an invention providing a data processor that can execute instructions of different formats utilizing different numbers of registers. Kishida, col. 3, lines 35-40. Kishida teaches that the instructions themselves contain information about their format. Kishida, col. 3, lines 46-47. This eliminates the need for a signal to switch between instruction formats. Kishida, col. 3, lines 38-40.

Kishida does not teach the invention of dependent claim 3 which recites "changing the value of a flag according to the content of an extended addressing register" in order to facilitate the saving of contextual data described in independent claim 1. Kishida does describe an extended memory register file, Kishida, col. 9, lines 65-67, accessed via an instruction format that specifies a larger number of registers, Kishida, col. 3, lines 43-45. However, the each instruction itself carries information about its format, about whether it is accessing the extended memory. Kishida, col. 3, lines 46-47. Kishida does not teach a flag or set of flags to indicate whether an extended addressing register is in use. Kishida discusses a special instruction that sends a "mode setting signal" used to switch between instruction formats, Kishida, col. 2, lines 35-48, but this instruction is not a "flag" in the sense of claim 3. Finally, at no point does Kishida discuss the problem of saving contextual data. For these reasons, claim 3 is not obvious in light of Kishida, Tremblay and the applicants' admitted prior art.

Claim 4 is dependent on claim 1, and so is non-obvious in view of Tremblay and admitted prior art based on the arguments made for claim 1 above, and is non-obvious further in view of Kishida for the reasons as given for claim 3 above.

Claims 11 and 12 are dependent on claim 9, and are non-obvious in view of Tremblay and admitted prior art for the same reasons as given for claim 1 above, and is non-obvious further in view of Kishida for the reasons as given for claim 3 above.

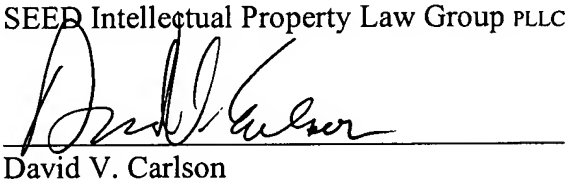
Claim 18 is dependent on claim 17, and is non-obvious in view of Tremblay and admitted prior art for the same reasons as given for claim 17 above, and is non-obvious further in view of Kishida based on the arguments made for claim 3 above.

The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC



David V. Carlson
Registration No. 31,153

DVC:lcs

701 Fifth Avenue, Suite 6300
Seattle, Washington 98104-7092
Phone: (206) 622-4900
Fax: (206) 682-6031

771052_1.DOC